CS 188 Summer 2020 — Introduction to Artificial Intelligence — Midterm 1

- You have approximately 110 minutes.
- The exam is open book, open calculator, and open notes.
- In the interest of fairness, we want everyone to have access to the same information. To that end, we will not be answering questions about the content or making clarifications.
- For multiple choice questions,
  - ☐ means mark **all options** that apply
  - ○ means mark a **single choice**

First name	
Last name	
SID	

For staff use only:

Q1.	Plants vs. Zombies	/22
Q2.	Climbing	/21
Q3.	RL: Stochastic Policies	/20
Q4.	Pacfriends Unite	/18
Q5.	Searching for a Problem...	/19
	Total	/100

THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [22 pts] Plants vs. Zombies

Dr. Zomboss wants your brain and is having zombies invade your backyard!

(a) [2 pts] You currently have the Peashooter **(P)**, but to defend yourself from the zombies, you need to purchase more plants from Crazy Dave! He has a plant lottery **(L)**, which gives either the Cherry Bomb **(C)**, the Wall-nut **(W)**, or the Snow Pea **(S)** with equal probability of each.

Assume rational preferences $P \succ W$, $L \succ P$, and $S \succ C$. Which of the following are guaranteed to be correct?

- ☐ $L \succ C$
- ☐ $C \sim L$
- ☐ $C \succ L$
- ☐ $P \succ C$
- ☐ $C \sim P$
- ☐ $C \succ P$
- ☐ $S \succ L$
- ◯ None of the Above

Now it's time to battle the zombies!

(b) You have M grass lanes and N types of plants. In each turn, Dr. Zomboss chooses a grass lane and puts a zombie on the right end of the lane. Then, you choose a plant to put on the left-most slot of that lane. You are trying to maximize utility and Dr. Zomboss is trying to minimize your utility.

Consider the game tree for one turn, and run alpha-beta pruning.

(i) [2 pts] For $M = 5$ and $N = 3$:

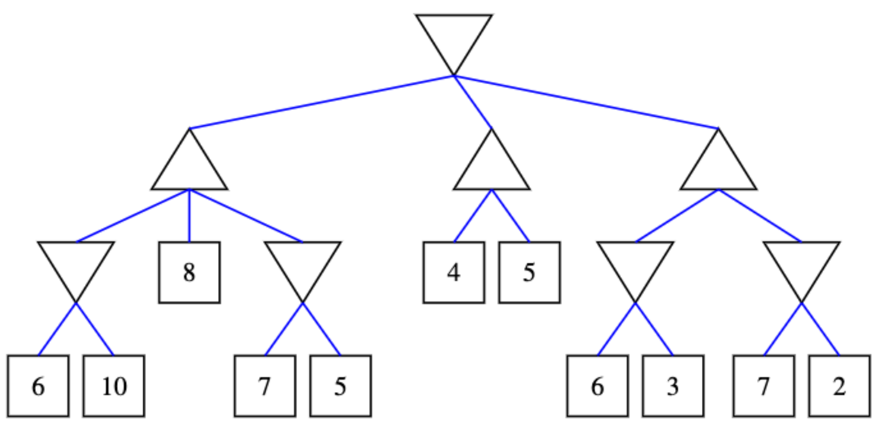
the maximum possible number of leaf nodes pruned = ☐.

the minimum possible number of leaf nodes pruned = ☐.

(ii) [1 pt] For $M = 6$ and $N = 10$:

the maximum possible number of leaf nodes pruned = ☐.

(c) Consider the following game tree:



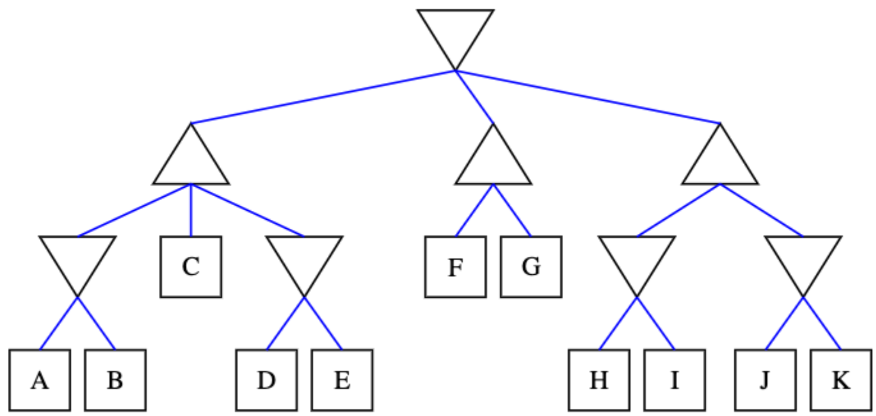
(i) [1 pt] Using alpha beta pruning, how many leaf nodes can we prune? Assume that branches are visited in left to right order.

.

(ii) [2 pts] If the branches of every minimizer node are reordered such that we prune the maximum number of leaf nodes, how many leaf nodes can now be pruned? Assume that children of maximizer nodes are visited from left to right, and that we are not pruning on equality.

.

(d) **Assume that we are not pruning on equality.** In this part, we have a fixed traversal order from left to right. We start with the tree in the previous part, and **shuffle the values of all the leaf nodes** such that we **check as few nodes as possible** without changing the structure of the tree. The initial ordering (as in the tree presented in the previous part) is 6, 10, 8, 7, 5, 4, 5, 6, 3, 7, 2, and the new ordering is $A, B, C, D, E, F, G, H, I, J, K$.



(i) [2 pts] For all possible new orderings, which of the results (root values) are possible?
☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 10

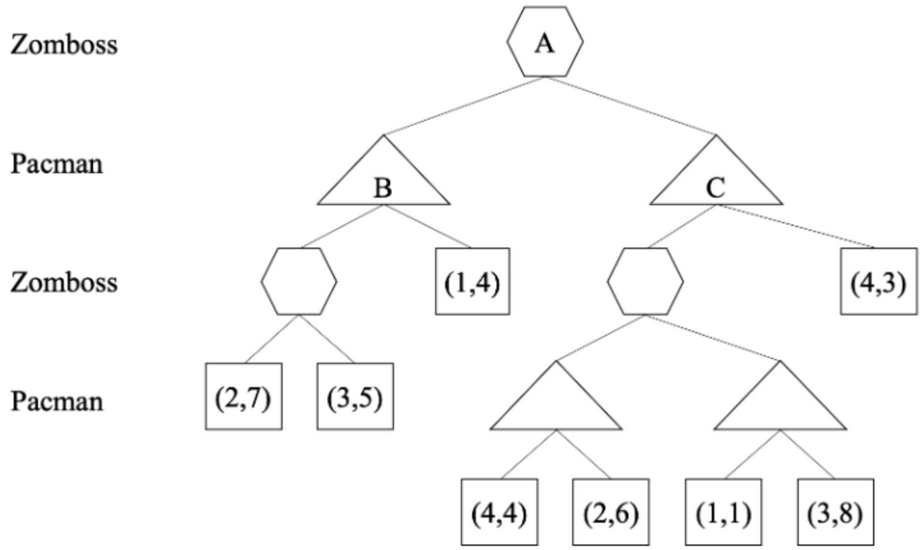
(ii) [3 pts] Suppose that for the new ordering, the value of the root is 4. Which of the following leaf nodes are guaranteed to have value ≤ 5?
☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G
☐ H ☐ I ☐ J ☐ K ○ None of the above

(iii) [3 pts] Suppose that for the new ordering, the value of the root is 5. Which of the following leaf nodes are guaranteed to have value > 5?
☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G
☐ H ☐ I ☐ J ☐ K ○ None of the above

(e) You are traveling and have asked Pacman to be in charge of your backyard, but Dr. Zomboss thinks you are the one at home, and is again coming for your brain!

Pacman knows Dr. Zomboss's utility, since you've been talking about it all the time, but **Dr. Zomboss doesn't know Pacman's utility**. However, Pacman does not care if zombies come into the house; he only wants more plants in the backyard. Both Pacman and Dr. Zomboss are playing optimally, trying to maximize their own utilities.

All nodes are in the format of (**Dr. Zomboss's utility, Pacman's utility**). Complete the tree below. In the case of a tie, choose the leftmost branch:



(i) [2 pts] Write node B as a tuple $(B_{Zomboss}, B_{Pacman})$.

$B_{Zomboss} =$ ☐ , $B_{Pacman} =$ ☐

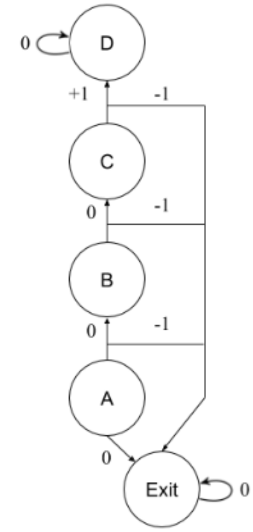
(ii) [2 pts] Write node C as a tuple $(C_{Zomboss}, C_{Pacman})$.

$C_{Zomboss} =$ ☐ , $C_{Pacman} =$ ☐

(iii) [2 pts] Write node A as a tuple $(A_{Zomboss}, A_{Pacman})$.

$A_{Zomboss} =$ ☐ , $A_{Pacman} =$ ☐

Q2. [21 pts] Climbing



Alice is deciding whether to spend a day at the Berkeley Ironworks to practice climbing. Since she recently took and mastered CS188, she decides to test her newfound knowledge to compute whether it will be optimal, from the point of view of her utility function, to go and practice. She decides to model her problem as the Markov Decision Process (MDP) shown on the right.

State A represents Alice's starting state, where Alice can either **stay** put on the ground and receive a reward of 0, or **climb** and advance to state B, also with a reward of 0. If Alice stays, she moves to a special "exit" state. If Alice begins climbing, she cannot stop, and must continue to advance to the next state until she reaches the top at state D, upon which she receives a reward of $+1$. However, while she is climbing (from A to B, from B to C, and from C to D), after attempting to move to the next state, there is a chance p_{fall} she may fall and receive a reward of -1, also causing Alice to enter the "exit" state.

Let's begin by assuming a falling probability of $p_{\text{fall}} = 0.1$ and a discount factor of $\gamma = 0.5$.

(a) [2 pts] (0.5 pt each) Assume we initialize the values of all states to 0. What is the value of each state after running a single iteration of value iteration?

- $V_1(A) =$ ______________
- $V_1(B) =$ ______________
- $V_1(C) =$ ______________
- $V_1(D) =$ ______________

(b) [4 pts] (1 pt each) What is the optimal value function at each state?

- $V^*(A) =$ ______________
- $V^*(B) =$ ______________
- $V^*(C) =$ ______________
- $V^*(D) =$ ______________

Alice wants to bring her friend Bob, but realizes that Bob might have a different skill level from her (p_{fall}) or might value time differently (γ). Therefore, Alice wants to estimate whether Bob will agree to **climb** or instead choose to **stay**.

(c) [3 pts] Assume $p_{\text{fall}} = 0.1$. For which of the following values of γ will Bob choose to **stay** from state A?

☐ 0.2
☐ 0.4
☐ 0.6
☐ 0.8

(d) [3 pts] Assume $\gamma = 0.5$. For which of the following values of p_{fall} will Bob choose to **stay** from state A?

☐ 0.05
☐ 0.1
☐ 0.15
☐ 0.2
☐ 0.25
☐ 0.3
☐ 0.35
☐ 0.4

Alice wants to try a different method for computing her total utility. Instead of using the discounted sum of rewards, Alice wishes to use the *average* reward she accumulates while climbing.

That is, rather than using the discounted sum of rewards to measure her total utility over an episode of k steps:

$$U^{\gamma}([r_0, \cdots, r_k]) = r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 r_3 + ... + \gamma^k r_k$$

She wishes to use the *average* utility:

$$\bar{U}([r_0, \cdots, r_k]) = \frac{1}{k+1}[r_0 + r_1 + r_2 + r_3 + ... + r_k]$$

(e) [3 pts] Assume $p_{\text{fall}} = 0.5$, and when we reach the exit state or the top state D, the episode terminates (rather than looping forever). What is Alice's expected *average* utility if she starts from state A, assuming Alice's policy is to always climb from A? *(Hint: the expectation is taken over randomness in falling (which can lead to some episodes being longer than others), and the average utility is taken over the length of an episode.)*

____

Partial credit (2 points) was given for not averaging over the timestep.

$$(1 - p_{\text{fall}})^3 - p_{\text{fall}} - (1 - p_{\text{fall}})p_{\text{fall}} - (1 - p_{\text{fall}})^2 p_{\text{fall}}$$

Plugging in $p_{\text{fall}} = 0.4$, we get a value of -0.568.

Plugging in $p_{\text{fall}} = 0.2$, we get a value of 0.024.

Plugging in $p_{\text{fall}} = 0.1$, we get a value of 0.458.

Recall that the total discounted utility can be written as a recursion.

$$U^\gamma([r_0, \cdots, r_k]) = \sum_{t=0}^{k} \gamma^t r_t = r_0 + \sum_{t=1}^{k} \gamma^t r_t = r_0 + \gamma U([r_1, \cdots, r_k])$$

This gives rise to the Bellman operator for policy evaluation, which computes this utility in expectation (the value function represents the total *expected* utility). Recall that in policy evaluation, we are given the value at iteration k, $V_k^\pi(s)$, and can use the following update to compute the value for iteration $k+1$, $V_{k+1}^\pi(s)$.

$$V_{k+1}^\pi(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_k^\pi(s')]$$

We are interested in deriving the corresponding update for the average utility case.

(f) [6 pts] (1 pt each; **(C)** and **(D)** worth 1.5 pts each)

Alice wishes to derive an algorithmic procedure for finding the average utility, $\bar{V}^\pi$. For each letter **(A), (B), (C), (D)** and **(E)**, fill in a single entry for the term corresponding to the correct equation to implement policy evaluation for the average utility case. Select one bubble per row to form the whole equation. *(Hint: try to think about how you can write down the average utility as a recursion, just like in the discounted utility case)*

$$\bar{V}_{k+1}^\pi(s) \leftarrow \textbf{(A)}\ \textbf{(B)}\ [\textbf{(C)} + \textbf{(D)}\ \bar{V}_k^\pi(s')] + \textbf{(E)}\ \bar{V}_k^\pi(s')$$

- **(A)** : ○ $\max_{\pi(s)}$ ○ $\sum_{s'}$ ○ $\max_{s'}$ ○ 1
- **(B)** : ○ $T(s, \pi(s), s')$ ○ $R(s, \pi(s), s')$ ○ 1
- **(C)** : ○ $R(s, \pi(s), s')$ ○ $\frac{1}{k+1} R(s, \pi(s), s')$ ○ $\frac{k}{k+1} R(s, \pi(s), s')$ ○ 0 ○ 1
- **(D)** : ○ γ ○ $\frac{1}{k+1}$ ○ $\frac{k}{k+1}$ ○ 0 ○ 1
- **(E)** : ○ γ ○ $\frac{1}{k+1}$ ○ $\frac{k}{k+1}$ ○ 0 ○ 1

Q3. [20 pts] RL: Stochastic Policies

Max and her friend Tubs are discussing solving Markov Decision Processes (MDP) and Reinforcement Learning (RL) problems. Max is wondering if she can arrive at the optimal value faster by changing the policy. Normally, in RL we use a **policy** $\pi(s)$ to choose an action at a given state: $\pi : s \mapsto a$. When estimating the value of a state, she knows that exploration and exploitation is an important trade-off. Max, knowing that we can incorporate exploration into Q-learning, wants to bring it to policy iteration. She decides to mimic epsilon-greedy exploration by making a **stochastic policy**. A stochastic policy, rather than return a single action, returns a probability distribution over potential actions.

$$\widetilde{\pi} : s \mapsto \mathcal{P}(a) \tag{1}$$

When taking an action from state s following policy $\widetilde{\pi}$, an agent samples an action at random according to the probability distribution $\widetilde{\pi}(s)$.

The initial MDP is below, and Max is annoyed that the policy she arrived at is clearly not optimal. Her friend Tubs suggests a change to stochastic policies, but she is skeptical that it'll help.

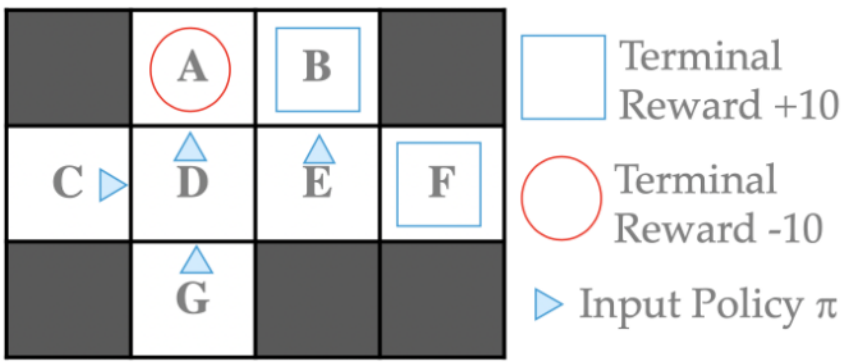


Figure 1: The MDP that Max is trying to solve.

state	policy
C	right
D	up
E	up
G	up

Table 1: Deterministic policy π.

state	$\mathcal{P}$(up)	$\mathcal{P}$(right)	$\mathcal{P}$(down)	$\mathcal{P}$(left)
C	0.1	0.8	0.1	0.0
D	0.7	0.1	0.1	0.1
E	0.4	0.4	0.1	0.1
G	0.8	0.1	0.0	0.1

Table 2: Stochastic policy, $\widetilde{\pi}$ for non-terminal states.

Above: first set of policies for Max.

Throughout this problem, we will operate with the following assumptions:
1) The discount factor, $\gamma = 1$.
2) The living reward is 0.
3) The transition functions, $\mathcal{T}(s, a, s')$*, are 'simple,' where the resulting motion is always in the direction of the action. For example,* $\mathcal{T}(C, right, D) = 1$, $\mathcal{T}(D, up, A) = 1$, $\mathcal{T}(E, down, E) = 1$.
4) Taking an action toward an invalid location will result in no movement, i.e.: the successor state for moving ***left*** *or* ***up*** *from state C in this MDP would also be state C.*
5) ***Arriving in a terminal state immediately grants the terminal reward and ends the episode.*** *For example,* $R(E, up, B) = 10$. *There is no "exit" action in this MDP.*

The Bellman equation for a fixed policy, shown below, may be useful during this problem.

$$V^{\pi}(s) = \sum_{s'} \mathcal{T}(s, \pi(s), s')\left[\mathcal{R}(s, \pi(s), s') + \gamma V^{\pi}(s')\right] \tag{2}$$

(a) [2 pts] Assuming $V_0^{\pi}(s) = 0$ for all states s, compute the value of the following states after two steps of **policy evaluation**, conditioned on the **deterministic policy**, π (shown in Table 1).

- $V_2^{\pi}(C) =$ ______________
- $V_2^{\pi}(D) =$ ______________
- $V_2^{\pi}(E) =$ ______________
- $V_2^{\pi}(G) =$ ______________

(b) [4 pts] Assuming $V_0^{\widetilde{\pi}}(s) = 0$ for all states s, compute the value of the following states after two steps of **policy evaluation**, conditioned on the **stochastic policy**, $\widetilde{\pi}$ (shown in Table 2). *(Hint: $V_k^{\widetilde{\pi}}(s)$ is the expected utility when taking at most k actions according to the policy $\widetilde{\pi}$.)*

- $V_2^{\widetilde{\pi}}(C) =$ ______________
- $V_2^{\widetilde{\pi}}(D) =$ ______________
- $V_2^{\widetilde{\pi}}(E) =$ ______________
- $V_2^{\widetilde{\pi}}(G) =$ ______________

(c) [2 pts] Given a deterministic, optimal policy, π^*, will changing to any probabilistic policy, $\widetilde{\pi}$, ever increase the **expected** reward an agent achieves?

○ Yes, always. ○ Sometimes. ○ No, never.

Now Max goes to a new MDP. She wants to learn the optimal policy and thinks that maybe stochastic policies will help in this case. For deterministic policies, we update the policy from a policy-conditioned value with *policy improvement*:

$$\pi_{i+1}(s) = \arg\max_a \sum_{s'} \mathcal{T}(s, a, s')\big[\mathcal{R}(s, a, s') + \gamma \mathbb{E}_{\pi(s)}[V^{\pi}(s')]\big] \tag{3}$$

We need to do something different for stochastic policies, so instead of returning the maximum action, we increase its probability. In the context of this problem, when we run policy improvement on the stochastic policy $\widetilde{\pi}$, we increase the probability of the best action by 0.3, and decrease all others by 0.1. (*In the case that an individual action is trying to decrease in probability below 0.0, its probability stays the same, and the change in probability for the best action is adjusted such that the distribution still sums to 1.0. For example: if exactly one action does not decrease, then the best action only increases by 0.2.*)

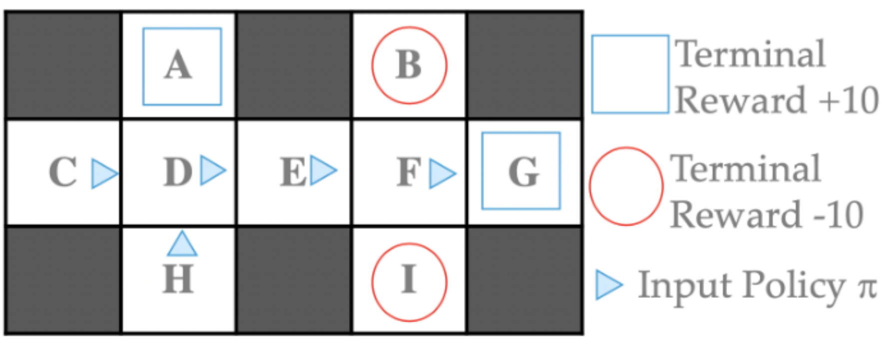


Figure 2: The MDP that Max is trying to solve.

In this part, the living reward is -1.

The other assumptions that we made previously still hold however:
1) The discount factor, $\gamma = 1$.
2) The transition functions, $\mathcal{T}(s, a, s')$, are 'simple,' where the resulting motion is always in the direction of the action. For example, $\mathcal{T}(C, right, D) = 1$, $\mathcal{T}(D, up, A) = 1$, $\mathcal{T}(E, down, E) = 1$.
*3) Taking an action toward an invalid location will result in no movement, i.e.: the successor state for moving **left** or **up** from state C in the current MDP would also be state C.*
*4) **Arriving in a terminal state immediately grants the terminal reward and ends the episode.** For example, R(F, right, G) = 10. There is no "exit" action in this MDP.*

state	policy
C	right
D	right
E	right
F	right
H	up

Table 3: Deterministic policy π.

state	$\mathcal{P}$(up)	$\mathcal{P}$(right)	$\mathcal{P}$(down)	$\mathcal{P}$(left)
C	0.1	0.7	0.1	0.1
D	0.1	0.7	0.1	0.1
E	0.1	0.3	0.1	0.5
F	0.1	0.5	0.1	0.3
H	0.7	0.1	0.1	0.1

Table 4: Stochastic policy, $\widetilde{\pi}$ for non-terminal states.

	sequence
1	C → D → E → F → G
2	H → D → E → F → G

Table 5: Deterministic policy roll-outs.

	sequence
1	C → D → A
2	H → D → E → F → E → F→ G
3	C → D → E → F → I

Table 6: Stochastic policy roll-outs.

(d) Compute the values of the listed states after TD-learning for the deterministic policy π in Table 3 and the stochastic policy $\widetilde{\pi}$ in Table 4. Use the provided episodes in Table 5 for π and the provided episodes in Table 6 for $\widetilde{\pi}$. ***In this part, the living reward is*** -1**.**

Hint: recall that TD-learning is performed after each transition **only**.

$V_0(s) = 0$ for all states (including terminal states).

$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha \cdot sample$, where $\alpha = .5$.

Note: in these parts it may be easiest to leave your answer as a fraction.

(i) [2 pts] Deterministic Policy

- State D ______________
- State E ______________

(ii) [2 pts] Stochastic Policy

- State D ______________
- State E ______________

(e) Compute the updated stochastic policy, using the value estimates V^{π} calculated from 3 episodes of TD learning (like in the previous subpart). ***In this part, the living reward is*** -1.

(i) [4 pts] State D.

- $\mathcal{P}$(up) = ______________
- $\mathcal{P}$(right) = ______________
- $\mathcal{P}$(down) = ______________
- $\mathcal{P}$(left) = ______________

(ii) [4 pts] State E.

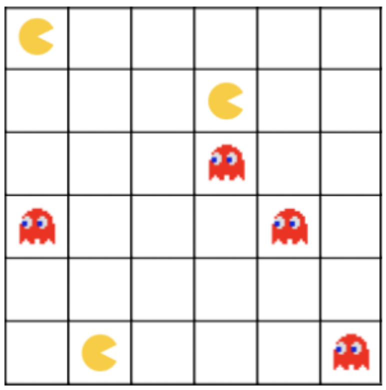
- $\mathcal{P}$(up) = ______________
- $\mathcal{P}$(right) = ______________
- $\mathcal{P}$(down) = ______________
- $\mathcal{P}$(left) = ______________

Q4. [18 pts] Pacfriends Unite

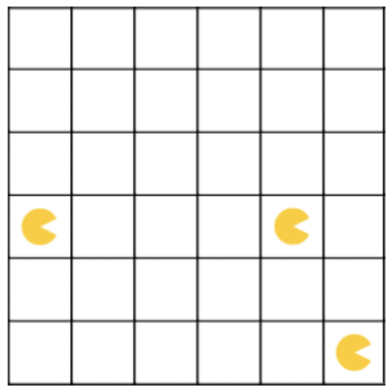
Pacman and his Pacfriends have decided to combine forces and go on the offensive, and are now chasing ghosts instead! In a grid of size M by N, Pacman and $P - 1$ of his Pacfriends are moving around to collectively eliminate **all** of the ghosts in the grid by stepping on the same square as each of them. Moving onto the same square as a ghost will eliminate it from the grid, and move the Pacman into that square.

Every turn, Pacman and his Pacfriends may choose one of the following four actions: *left, right, up, down*, but may not collide with each other. In other words, any action that would result in two or more Pacmen occupying the same square will result in no movement for either Pacman or the Pacfriends. Additionally, Pacman and his Pacfriends are **indistinguishable** from each other. There are a total of G ghosts, which are indistinguishable from each other, and cannot move.

Treating this as a search problem, we consider each configuration of the grid to be a state, and the goal state to be the configuration where **all** of the ghosts have been eliminated from the board. Below is an example starting state, as well as an example goal state:



(a) Possible Start State



(b) Possible Goal State

Assume each of the following subparts are **independent** from each other. **Also assume that regardless of how many Pacmen move in one turn, the total cost of moving is still 1.**

(a) [8 pts] Suppose that Pacman has no Pacfriends, so $P = 1$.

(i) [2 pts] What is the size of the minimal state space representation given this condition? Recall that $P = 1$.

○ MN ○ $(MN)^G$ ○ 2^{MN} ○ $G(2)^{MN}$

○ MNG ○ $(MN)^{G+1}$ ○ 2^{MN+G} ○ $MN(2)^G$

For each of the following heuristics, select whether the heuristic is only admissible, only consistent, neither, or both. Recall that $P = 1$.

(ii) [2 pts] $h(n)$ = the sum of the Manhattan distances from Pacman to every ghost.

○ only admissible ○ only consistent ○ neither ○ both

(iii) [2 pts] $h(n)$ = the number of ghosts times the maximum Manhattan distance between Pacman and any of the ghosts.

○ only admissible ○ only consistent ○ neither ○ both

(iv) [2 pts] $h(n)$ = the number of remaining ghosts.

○ only admissible ○ only consistent ○ neither ○ both

(b) [10 pts] Suppose that Pacman has exactly one less Pacfriend than there are number of ghosts; therefore $P = G$. Recall that Pacman and his Pacfriends are indistinguishable from each other.

(i) [2 pts] What is the size of the minimal state space representation given this condition? Recall that $P = G$.

○ MNP
○ $MNGP$
○ $(MN)^G$
○ $(MN)^{(G+P)}$
○ $(MN)^P 2^G$
○ $(MN)^G P$
○ $(MN)^{G+1}$
○ $(MN)^{(G+1)P}$
○ $\binom{MN}{P}$
○ $\binom{MN}{P} 2^G$
○ $\binom{MN}{P}(MN)^G$
○ $\binom{MN}{P}\binom{MN}{G}$
○ 2^{MN}
○ 2^{MN+G+P}
○ $GP(2)^{MN}$

For each of the following heuristics, select whether the heuristic is only admissible, only consistent, neither, or both. Recall that $P = G$.

(ii) [2 pts] $h(n)$ = the largest of the Manhattan distances between each Pacman and its closest ghost.

○ only admissible ○ only consistent ○ neither ○ both

(iii) [2 pts] $h(n)$ = the smallest of the Manhattan distances between each Pacman and its closest ghost.

○ only admissible ○ only consistent ○ neither ○ both

(iv) [2 pts] $h(n)$ = the number of remaining ghosts.

○ admissible ○ only consistent ○ neither ○ both

(v) [2 pts] $h(n) = \frac{\text{number of remaining ghosts}}{P}$.

○ only admissible ○ only consistent ○ neither ○ both

Q5. [?? pts] Searching for a Problem...

(a) [2 pts] Mark the statement(s) below that are correct. Assume all search problems are on finite graphs for this problem.

- ☐ DFS graph search is guaranteed to find a solution (if one exists).
- ☐ A* graph search is guaranteed to find a solution (if one exists) with any admissible heuristic.
- ☐ UCS does not always find the optimal solution.
- ☐ There are cases where BFS tree search finds a solution but DFS tree search doesn't.
- ☐ There are cases where DFS tree search finds a solution but BFS tree search doesn't.

(b) Turtle Tee is crawling within an $M \times N$ coral habitat, located in the middle of an $X \times Y$ area. Tee knows that his dinner has been hidden by a naughty fish within the coral habitat, and Tee needs to find it to stay alive. The food will be consumed once (and only once) when Turtle Tee reaches its position. Even though Tee cannot leave the coral habitat, he is determined to survive and would like your help!

(i) [3 pts] Please complete the following expression that evaluates to the size of the minimum state space by filling in the blanks. For example, writing 0 in all the blanks will result in taking all of these variables to the 0th power:

$$2^{(a)} \cdot M^{(b)} \cdot N^{(c)} \cdot X^{(d)} \cdot Y^{(e)}$$

(a) ☐ **(b)** ☐ **(c)** ☐ **(d)** ☐ **(e)** ☐

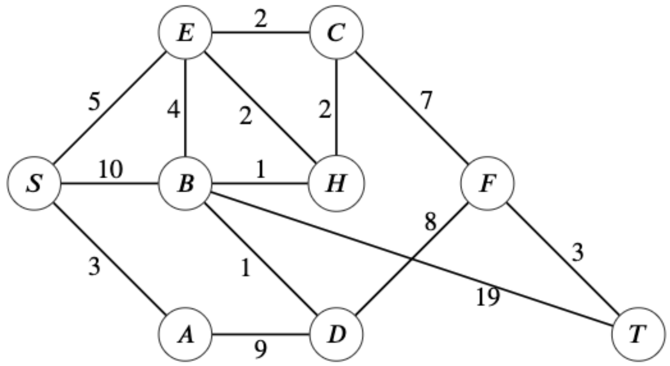
(ii) [4 pts] We now know that the naughty fish also set up $K > 10$ traps inside the coral region (many could be overlapping, and could be right on top of the food Tee is looking for). **Traps are visible, stationary, and each trap will trigger every time it is stepped on.** If Tee steps on 10 traps before he finds his dinner, then he would be so severely injured that he couldn't move anymore, and would then starve! We still want to make sure that the food can only be consumed once (when and only when Turtle Tee steps on that square).

Please complete the following expression that evaluates to the size of the minimum state space where **(g)** represents a constant and A **represents the size of the correct minimal state space from the previous part:**

$$A \cdot 2^{\mathbf{(a)}} \cdot M^{\mathbf{(b)}} \cdot N^{\mathbf{(c)}} \cdot X^{\mathbf{(d)}} \cdot Y^{\mathbf{(e)}} \cdot K^{\mathbf{(f)}} \cdot \mathbf{(g)}$$

(a) ☐ **(b)** ☐ **(c)** ☐ **(d)** ☐ **(e)** ☐ **(f)** ☐ **(g)** ☐

(c) Suppose the numbers in the following graph represent the cost of traversing its corresponding edge. In this problem, we use a priority queue representation for our fringe in **all** of the following search algorithms, and break ties alphabetically for items at the same priority. We start from node S (leftmost), and the goal is to find node T (rightmost). Also provided are the heuristic values for each node.



Node	S	A	B	C	D	E	F	H	T
h(Node)	19	15	18	9	10	12	3	2	1

For the following parts, list the order of nodes expanded using the specified search algorithm separated by commas (no spaces), e.g., an example ordering is S,A,D,F,T.

(i) [2 pts] What is the order of node expansion using BFS graph search?

(ii) [2 pts] What is the order of node expansion using DFS graph search?

(iii) [2 pts] What is the order of node expansion using UCS graph search?

(iv) [2 pts] What is the order of node expansion using A* graph search?

(v) [2 pts] Using the heuristics table associated with the graph, select the node(s) that have inadmissible heuristic value(s):

☐ S
☐ A
☐ B
☐ C
☐ D
☐ E
☐ F
☐ H
☐ T